TabEL: Entity Linking in Web Tables

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey

Northwestern University Evanston IL 60201, USA,
{csbhagav, nor.thanapon}@u.northwestern.edu,
ddowney@eecs.northwestern.edu

Abstract. Web tables form a valuable source of relational data. The Web contains an estimated 154 million HTML tables of relational data, with Wikipedia alone containing 1.6 million high-quality tables. Extracting the semantics of Web tables to produce machine-understandable knowledge has become an active area of research.

A key step in extracting the semantics of Web content is *entity linking* (EL): the task of mapping a phrase in text to its referent entity in a knowledge base (KB). In this paper we present *TabEL*, a new EL system for Web tables. *TabEL* differs from previous work by weakening the assumption that the semantics of a table can be mapped to pre-defined types and relations found in the target KB. Instead, *TabEL* enforces soft constraints in the form of a graphical model that assigns higher likelihood to sets of entities that tend to co-occur in Wikipedia documents and tables. In experiments, *TabEL* significantly reduces error when compared to current state-of-the-art table EL systems, including a 75% error reduction on Wikipedia tables and a 60% error reduction on Web tables. We also make our parsed Wikipedia table corpus and test datasets publicly available for future work.

Keywords: Web tables, Entity Linking, Named Entity Disambiguation, Graphical Models

1 Introduction

Web tables, or HTML tables on the Web, are a valuable source of relational data and an important input for information extraction (IE) systems. It is estimated that out of a total of 14.1 billion tables on the Web, 154 million tables contain relational data [1] and Wikipedia alone is the source of nearly 1.6 million relational tables. Unlike text, a single relational table contains a high-quality set of relation instances, along with associated metadata (in the form of column headers). The wealth and utility of relational tables on the Web has made *semantic interpretation* of tables, i.e. the task of converting Web tables into machine-understandable knowledge, an active area of research [2–12].

A key step in extracting the semantics of Web content is entity linking (EL): the task of mapping phrases of text to their referent entities in a given Knowledge Base (KB). For example, in Table 1, the EL task is to link “Chicago” in the second column to its corresponding entity `Chicago` (the city) in a KB, e.g. YAGO [13].[1] Polysemy of

[1] `https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entityIn=%3CChicago%3E`

phrases is the main challenge for EL systems. An EL system must disambiguate each given phrase utilizing clues from surrounding content, called the *context* of the phrase. In Table 1, the phrase “New York” occurs multiple times, but it is evident from context that it refers to the city in the second column and to the state in the third column.

We present *TabEL*, a system that performs the *Entity Linking* task on phrases in cells of Web tables. Existing table semantic interpretation systems typically employ graphical models to jointly model three semantic interpretation tasks: *entity linking*, *column type identification* and *relation extraction* from tables (detailed in Section 2) [4,6,7,12]. Such joint models are based on a strong assumption that the column types and relations expressed in a table can be mapped to pre-defined types and relations in the target KB. While the type and relation information conveyed by the structure of tables are valuable clues for the EL task, relying on a strict mapping into a KB is prone to errors as KBs can be incomplete or noisy.

Table 1: Table containing a list of tallest buildings in the U.S. and the city and state that they are located in. Underlines represent an existing reference to an entity in a KB.

Building Name	**City**	**State**
One WTC	New York	New York
Willis Tower	Chicago	Illinois
⋮	⋮	⋮
MetLife Tower	New York	New York

In this paper, we investigate an alternative to the strict mapping into a KB. *TabEL* incorporates type and relation information through a graphical model of soft constraints. The constraints encode a preference for sets of referent entities that are “coherent”, in that pairs of entities in the set tend to co-occur in Wikipedia documents and tables. Although our graphical model is densely connected (see Section 3), we show in experiments that we can tractably arrive at disambiguations using the Iterative Classification Algorithm (ICA) [14]. In experiments, we show that *TabEL* is more accurate than previous work, reducing error over the benchmark system [4] by ∼60% on Web tables. *TabEL* performs particularly well on Wikipedia tables and reduces error over previous work by ∼75%. In ablation studies, we analyze the impact of *TabEL*’s components on accuracy and demonstrate that our features result in an improvement of ∼12% over a system that chooses the most usual meaning of a phrase as its referent entity.

Finally, we release our table corpus containing more than 1.6 million tables from Wikipedia. We also make datasets of entity-annotated Wikipedia and Web tables publicly available for future table EL systems.[2]

2 Preliminaries

The general task of semantic interpretation of tables takes as input a table and a reference Knowledge Base (KB), and typically includes the following sub-tasks:

1. *Entity linking* (EL): the task of finding phrases of text, called *mentions*, in cells and associating each with its referent entity
2. *Column type identification*: the task of associating a column in a table with the KB type of entities it contains

[2] `http://websail-fe.cs.northwestern.edu/TabEL/`

3. *Relation extraction*: the task of associating a pair of columns in a table with the KB relation that holds between each pair of entities in a given row of the columns

The referent entities, types and relations are all grounded in the given KB. As a concrete example, given Table 1 and the YAGO [13] KB, the entity linking task would include linking "Chicago" to the entity `Chicago` in the KB.[3] Type identification would include associating the second column to the `City` type in the KB.[4] The relation extraction task would include identifying the relation `isLocatedIn` between entities `Willis_Tower` and `Chicago`.[5]

In this paper, we focus on just the first semantic interpretation task, entity linking. We now formally define the EL task for tables. We also introduce notation that will be used in the rest of this paper.

Formal Definition

A *potential mention* is a phrase in text whose referent entity in the given KB is unknown. We denote a potential mention for a phrase s as $m_{s,\mathbf{?}}$ (where ? denotes an unknown entity). An *annotated mention*, on the other hand, is a phrase whose referent entity is known and is denoted by $m_{s,e}$, where s is the phrase of text whose referent entity is e.

A *table* from the Web is represented as a matrix, T, of cells containing r rows and c columns. Tables that use row- and column-spans can be easily normalized into an $r \times c$ matrix by duplicating cells. $T[i, j]$ represents the cell in the i^{th} row and j^{th} column of T.

Task: Given a table T and a KB $\mathcal{K}$ of entities, the *entity linking* task is to identify and link each potential mention in cells of T to its referent entity $e \in \mathcal{K}$.

3 System Description

Given a table T and a KB $\mathcal{K}$, *TabEL* performs the EL task in three steps:

1. *mention identification:* identifies each potential mention, $m_{s,\mathbf{?}}$, in cells of T
2. *entity candidate generation:* for each potential mention $m_{s,\mathbf{?}}$, identifies a set of candidate entities, $\mathcal{C}(m_{s,\mathbf{?}})$ - a subset of entities in $\mathcal{K}$ that are possible referents of $m_{s,\mathbf{?}}$
3. *disambiguation:* for each potential mention $m_{s,\mathbf{?}}$, chooses an entity $e \in \mathcal{C}(m_{s,\mathbf{?}})$ (its candidate set), as the referent entity of $m_{s,\mathbf{?}}$, based on its context.

TabEL uses a supervised learning approach, and uses annotated mentions in tables to train its components. Like most EL systems, *TabEL* also relies on a prior estimate that a given string s refers to a particular entity e, i.e. $P(e|s)$. As in previous work [15], we estimate this distribution $P(e|s)$ from hyperlinks on the Web and in Wikipedia, as described in Section 4 .

[3] `https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entityIn=%3CChicago%3E`

[4] `https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entity=%3Cwordnet_city_108524735%3E`

[5] `https://gate.d5.mpi-inf.mpg.de/webyagospotlx/WebInterface?LO1=%3CWillis_Tower%3E&LOR=%3CisLocatedIn%3E&LO2=%3CChicago%3E`

While we use YAGO as our knowledge base in our experiments, our approach is general and can use any KB, given some labeled examples for that KB and a suitable entity-similarity measure that we use in our system.

3.1 Mention Identification

The first step for any EL system is to find potential mentions that can be linked to their referent entities in $\mathcal{K}$. Given the text content, t_q, of each cell of the input table, *TabEL* identifies as a potential mention the longest phrase, s of t_q that has non-zero probability in $P(e|s)$ for some e. If the length of s is less than the length of t_q, *TabEL* finds the longest phrase starting after s and so on. For example, for a cell with text "Barack Obama & Mitt Romney", *TabEL* finds two potential mentions: one for "Barack Obama" and one for "Mitt Romney".

3.2 Candidate Generation

For each potential mention, $m_{s,?}$, *TabEL* sets the set of candidate entities $\mathcal{C}(m_{s,?})$ for the mention to be all those e for which $P(e|s)$ has non-zero probability, i.e. $\mathcal{C}(m_{s,?}) = \{e|P(e|s) > 0\}$. For example, the candidate entity set for the phrase "Chicago" would contain the entities `Chicago`, `Chicago_Bulls`, `Chicago_(1927_film)`, etc.

3.3 Disambiguation

Our disambiguation technique is based on the assumption that entities in a given row or column tend to be related. As we show in our experiments, when disambiguating multiple cells of a table, we can achieve higher accuracy by preferring sets of disambiguations that are coherent (i.e. sets composed of related entities). To exploit this fact, we utilize a collective classification technique in which soft constraints encourage disambiguations of mentions in the same row and column to be related to one another. The disambiguations in a given table are optimized jointly, to arrive at a globally coherent set of entities.

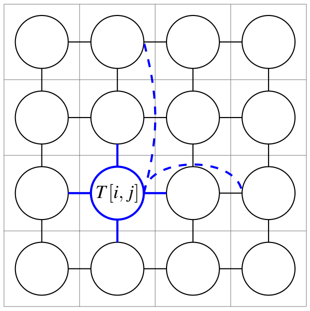

Fig. 1: Graphical Model used for disambiguation. Circles represent variables and edges represent their dependencies. For brevity, we show non-adjacent dependencies only for the cell $T[i,j]$

In the disambiguation step, an EL system needs to choose an entity from the candidate set $\mathcal{C}(m_{s,?})$ as the referent entity of a given mention $m_{s,?}$. We represent a table, T, as a graphical model in which each potential mention is associated with a discrete random variable, whose possible values are its candidate entities. Each variable has a direct dependency with all other variables in its row and column. The model can be drawn as a Markov Network, shown in Figure 1, in which each row (and each column) forms a fully-interconnected clique.

Our graphical model is much more densely connected than the models used in previous work on this task [4,7]. However, we find that an iterative, approximate inference

Algorithm 1 ICA for Disambiguation in *TabEL*

```
 1: function TabEL-ICA($\mathcal{M}_{LR}$, $T$, maxIter)      ▷ $\mathcal{M}_{LR}$: Local Disambiguation Model
                                                             ▷ $T$: Input Table
                                  ▷ maxIter: Maximum number of inference iterations
 2:     for all $m_{s,?} \in T$ do
 3:         $m_{s,?} \leftarrow m_{s,e_0}$                   ▷ where $e_0 \leftarrow \mathcal{M}_{LR}(\mathcal{C}(m_{s,?}))$
 4:     end for
 5:     $k \leftarrow 1$
 6:     do
 7:         for all $m_{s,?} \in T$ do
 8:                ▷ Re-calculate features according to current assignment to other variables
 9:             $reCalculateFeatures(m_{s,?})$
10:         end for
11:         $hasChange \leftarrow False$
12:         for all $m_{s,?} \in T$ do
13:             $m_{s,e_{k-1}} \leftarrow m_{s,e_k}$          ▷ Re-assign value. $e_k \leftarrow \mathcal{M}_{LR}(\mathcal{C}(m_{s,e_{k-1}}))$
14:             if $e_{k-1} == e_k$ then
15:                 $hasChange \leftarrow True$
16:             end if
17:         end for
18:     while $hasChange$ AND $k < maxIter$
19: end function
```

approach is tractable for the model. *TabEL* uses the Iterative Classification Algorithm (ICA) [14] to collectively disambiguate all mentions in a given table. ICA is an iterative inference method which greedily re-assigns each variable to its maximum-likelihood value, conditioned on the current values of other variables. In each iteration, we compute the maximum-likelihood value for each variable using a trained *local classifier*, $\mathcal{M}_{LR}$, which takes the form of a logistic-regression-based ranking model. Algorithm 1 shows how ICA performs iterative inference over the graphical model to find a high-likelihood set of referent entities for all mentions in a given test table. The method initializes each mention with an entity using $\mathcal{M}_{LR}$ (lines 2 to 4) and then iteratively re-computes features and assignments (lines 6 to 18) until there is no change in assignment for any mention or the maximum iteration limit is reached.

$\mathcal{M}_{LR}$ ranks the candidate entities for a given mention, based on a set of features computed from the current settings of the other variables. The local model $\mathcal{M}_{LR}$ is trained in advance on a set of annotated mentions. $\mathcal{M}_{LR}$ utilizes the following groups of features:

Prior probability features, $P(e|s)$ are estimated from hyperlinks on the Web and in Wikipedia. For example, the phrase “Chicago” appears 16,884 times as an anchor text in Wikipedia. It links to one of 289 distinct pages including the city, the movie, the music band etc. But the string “Chicago” most likely refers to the city ($P(\texttt{Chicago\_City}$ $|“\textit{Chicago}'') = 0.80$). For each distinct source of hyperlinks, we compute features for both case-sensitive and case-insensitive matching of the phrase. In addition, we include the averages of case-sensitive and case-insensitive probability estimates across all sources.

Semantic relatedness (SR) features are used to measure the coherence between a candidate entity and other entities in the table. *TabEL* has three SR based features: average SR between a candidate entity and all entities in the mention's 1) row, 2) column, and 3) context i.e. row and column.

In *TabEL*, we use SR defined between a pair of Wikipedia pages based on their in-link and out-link overlap. We use the SR implementation from Hecht et al. [16]. This is a modified version of Milne-Witten Semantic Relatedness measure [17] in which the links in the first paragraph of a Wikipedia page are considered more important than other links when calculating relatedness. The average SR value between a candidate entity and entities in a mention's context is an important feature for the EL task in tables, as shown by our experiments in Section 5. In the special case of applying *TabEL* to Wikipedia tables, we also include a feature for relatedness between the candidate entity and the Wikipedia page containing the table.

Mention-Entity Similarity features capture the similarity between the *context* of a potential mention and the *context-representation* of each of its candidate entities. We define the context of a mention as the contents of the cells in its row and column. The context-representation of an entity is the aggregation of the contexts in which it occurs in the training data.

For example, $m_{Chicago,?}$ is a potential mention in the cell $T[2,2]$ in Table 1. The highlighted column is referred to as the column context of the mention, denoted by $\mathcal{X}^C(T[2,2])$. Similarly, the highlighted row is referred to as the row-context and denoted by $\mathcal{X}^R(T[2,2])$. Consider the entity `New_York_City` in $T[1,2]$ in Table 1. Its context contains entities `Chicago`, `One_World_Trade_Center`, `MetLife_Tower` etc. To construct a context-representation for `New_York_City`, we aggregate the contexts of all mentions in our table corpus that link to `New_York_City`.

In general, the row and column contexts of a mention in cell $T[i,j]$ are given by:

$$\mathcal{X}^R(T[i,j]) = T[i,\cdot] \setminus T[i,j]$$
$$\mathcal{X}^C(T[i,j]) = T[\cdot,j] \setminus T[i,j]$$

where T[i,·] refers to the cells in the i^{th} row and T[·, j] refers to cells in the j^{th} column. $\mathcal{X}^R_W(T[i,j])$ denotes a multiset of word tokens found in $\mathcal{X}^R(T[i,j])$. $\mathcal{X}^R_E(T[i,j])$ denotes a multiset of entities found in $\mathcal{X}^R(T[i,j])$. Similarly, we define $\mathcal{X}^C_W(T[i,j])$ and $\mathcal{X}^C_E(T[i,j])$ to denote multisets of word tokens and entities found in $\mathcal{X}^C(T[i,j])$.

The context-representation of an entity can be derived from a corpus of tables, $\mathcal{T}$ with annotated mentions. We define two kinds of context-representations for an entity: 1) word-context-representation, $\mathcal{R}_W(e)$ is an aggregation of words and their frequencies from the contexts of all cells in $\mathcal{T}$ which contain a reference to e. 2) entity-context-representation $\mathcal{R}_E(e)$ is a similar aggregation of entities and their frequencies. Formally,

$$\mathcal{R}_W(e) = \biguplus_{T\in\mathcal{T}}\left(\mathcal{X}^R_W(T[i,j]) \uplus \mathcal{X}^C_W(T[i,j])\right)$$
$$\mathcal{R}_E(e) = \biguplus_{T\in\mathcal{T}}\left(\mathcal{X}^R_E(T[i,j]) \uplus \mathcal{X}^C_E(T[i,j])\right)$$

where, $m_{\cdot,e} \in \mathrm{T}[i,j]$, i.e. the cell $T[i,j]$ contains a mention whose target entity is e and $\uplus$ denotes a multiset union.

TabEL uses the following six features based on similarity between a mention's contexts and a candidate entity's context-representations.

Text-context similarity features	**Entity-context similarity features**
$S_C\Big(\mathcal{X}_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$
$S_C\Big(\mathcal{X}^R_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}^R_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$
$S_C\Big(\mathcal{X}^C_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}^C_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$

where, S_C denotes cosine similarity between the two multisets. We weight the multiplicity of the words and entities in these multisets by their Residual IDF (r-idf) values [18], which we pre-computed for our corpus.

Existing Link features are related to mentions that are already linked to their referent entity in the input table. We include two boolean features in our system. The first feature captures whether there is an existing mention in the context of $m_{s,?}$ with the same surface s that links to the candidate entity. The second feature captures whether the candidate is linked from a surface s' different from s in the input table.

Surface features are related to the phrase, s of the potential mention $m_{s,?}$. We have two boolean features. The first feature is true if s is the only text content in its cell, false otherwise. The second feature is true if s exactly matches the name of an entity in the input KB, $\mathcal{K}$.

4 Implementation

Table Corpus: Our dataset of tables $\mathcal{T}$ has 1.6 million Wikipedia tables. We extracted all HTML tables from Wikipedia which had the class attribute "wikitable" (used to easily identify data tables) from the November 2013 XML dump of English Wikipedia using the Sweble parser [19]. [6] As described in Section 2, all HTML tables are represented as an $r \times c$ matrix of cells. Tables in $\mathcal{T}$ contain ~ 30 million hyperlinks in all. 75% of these hyperlinks are used to build other resources described below. The other 25% are exclusively used for training, validation and testing of the local disambiguation model $\mathcal{M}_{LR}$, described in Section 3.3.

Knowledge Base of Entities: We use YAGO, which contains more than 2.8 million entities, as our reference KB, $\mathcal{K}$. *TabEL* links mentions to one of these 2.8 million entities in $\mathcal{K}$. YAGO contains a bi-directional mapping between Wikipedia pages and its entities. We exploit this mapping to identify the YAGO entity of the targets of hyperlinks in $\mathcal{T}$.

Source of Annotated Mentions: As mentioned in Section 3, we utilize a dataset of annotated mentions to train the $\mathcal{M}_{LR}$ model and to construct context representations described in Section 3.3. As explained above, pages on Wikipedia can be easily mapped to entities in the YAGO knowledge base. Thus, annotated mentions can be obtained from

[6] `http://dumps.wikimedia.org/enwiki/20131104/`

hyperlinks on the Web and Wikipedia by considering the anchor text as the phrase and the link target as its referent entity in the KB, $\mathcal{K}$.

To reliably estimate the probability that a surface s refers to an entity e, we use hyperlinks from both the Web and Wikipedia. The Google Cross-Lingual Dictionary for English Wikipedia Concepts described by Spitkovsky et al. [20] contains a dataset of all hyperlinks on the Web which link to a page in Wikipedia. We augmented this with hyperlinks obtained from Wikipedia. We mined over 100 million hyperlinks from the Web and Wikipedia and obtained a large dataset of annotated mentions. With the availability of high quality resources such as the Google Cross-Lingual Dictionary, EL systems that rely *only* on prior probability, $P(e|s)$, can still perform very well. We include a system, $\textit{TabEL}_{prior}$ in our experiments which disambiguates a potential mention by choosing the most frequently linked entity, for a given phrase, as its referent entity.

5 Experiments

In this section, we evaluate the accuracy of *TabEL* and compare with previous work on both (i) Web tables and (ii) Wikipedia tables. In an ablation study we evaluate the utility of each group of features employed in *TabEL*, and establish the importance of features that are based on entity co-occurrence. We show the effectiveness of the collective inference method, ICA, in improving the accuracy of *TabEL*.

Evaluation Metric: *TabEL* performs disambiguation on all test mentions and always chooses an entity that exists in the given KB. Thus, following previous work on table EL, we use accuracy as our main metric for evaluation and comparison with other table EL systems. We define accuracy as the fraction of test set mentions that an EL system links correctly. For comparison with text EL systems, we use the macro-averaged precision, recall and F1 metrics, which are popularly used for the text EL task.

5.1 Web tables

To evaluate the performance of *TabEL* on Web tables, we use the WEB_MANUAL dataset from previous work by Limaye et al. [4], both with and without corrections as described below. The WEB_MANUAL dataset consists of more than 9,000 test mentions from 428 tables from the Web. Using methods from Gupta et al. [21], this dataset was originally created (in [4]) by finding Web tables similar to a seed set of 36 non-infobox tables from Wikipedia. Out of the 9,036 test mentions in the WEB_MANUAL dataset, we found that around 5% of the gold annotations were erroneous. Mulwad et al. [7] have also noted errors in the gold annotations of this dataset, but left corrections to future work. We re-labeled these erroneous mentions and created a new dataset, WEB_MANUAL-FIXED, of Web tables with corrected annotations.

Table 2 shows the accuracy of $\textit{TabEL}_{prior}$ and *TabEL* on the fixed dataset, WEB_MANUAL-FIXED. For completion and comparison with previous work, we also show the accuracy of our system on the original WEB_MANUAL dataset. *TabEL* outperforms previous work on the WEB_MANUAL dataset and $\textit{TabEL}_{prior}$ on the WEB_MANUAL-FIXED dataset. A list of errors we found in the gold annotations in WEB_MANUAL and the re-annotated dataset WEB_MANUAL-FIXED are available on our project web page.

Table 2: Accuracy comparison between previous work and *TabEL* on Web and Wikipedia tables datasets

Dataset	**Limaye et al.** [4]	***TabEL**$_{prior}$*	***TabEL***
WEB_MANUAL	81.37	84.41	**89.41**
WEB_MANUAL- FIXED	-	87.56	**92.94**
WIKI_LINKS	84.28	91.27	**97.16**
WIKI_LINKS- RANDOM	-	87.83	**96.17**

5.2 Wikipedia Tables

In Table 2 we show the performance of *TabEL* on two datasets derived from Wikipedia. We adopted the WIKI_LINKS dataset from previous work by Limaye et al. [4], which consists of more than 140,000 test mentions from around 3,000 tables from Wikipedia. *TabEL* outperforms previous work by reducing the error on the WIKI_LINKS dataset by more than 75%.

We evaluate on a second dataset, WIKI_LINKS-RANDOM, of Wikipedia tables in an attempt to provide a more comprehensive measure of performance. The WIKI_LINKS dataset, from Limaye et. al [4], was originally constructed by choosing Wikipedia tables which contained links in at least 90% of their cells. We believe that this dataset is possibly biased as the high density of links in the tables suggests that the tables are important and probably contain commonly known entities in their cells. This bias is evident from the contrast in performance of *TabEL$_{prior}$* on the WIKI_LINKS and WIKI_LINKS-RANDOM datasets. The *TabEL$_{prior}$* system, which selects the most common referent entity for a given text mention, performs much better on WIKI_LINKS compared to WIKI_LINKS-RANDOM. Thus, we created the WIKI_LINKS-RANDOM dataset containing randomly selected Wikipedia tables, irrespective of the density of existing links in the tables. WIKI_LINKS-RANDOM consists of around 50,000 test mentions from around 3000 tables randomly drawn from Wikipedia. Each existing link in a table is used as a test mention, with its target entity treated as a gold annotation.

Table 2 shows that *TabEL* achieves very high accuracy on both WIKI_LINKS and WIKI_LINKS-RANDOM datasets. Performing table EL on Wikipedia tables at high level of accuracy is important as many systems utilize links in Wikipedia tables to create RDF triples or to support table search systems (see Section 7).

5.3 Disambiguating Missing Wikipedia Links

An interesting variation of the EL task for tables is to identify and disambiguate unlinked mentions to entities in a Wikipedia table - while *retaining* existing links, unlike the experiments above in Section 5.2 in which all existing links were removed. To evaluate *TabEL* on this task, we created a dataset, TABEL_35K, containing 35,000 randomly selected annotated mentions in Wikipedia. These mentions are not used in estimating prior probabilities and in building context representations. *TabEL* performs particularly well on this task and its accuracy on this dataset is 98.38%, while the accuracy of *TabEL$_{prior}$* is 88.13%. Interestingly, we found that 16% of the errors made by *TabEL* on this dataset are actually not errors in *TabEL*, but instead errors in the hyperlinks within Wikipedia. Another 22% of the errors are cases for which both the gold annotation and

TabEL's annotation can be considered correct for the mention. Details of all errors made by *TabEL* on this dataset can be found linked from our project page.

5.4 Comparison with other Table EL systems

Zhang et al. [12] introduced a table EL system that jointly performs the three semantic interpretation tasks. Direct comparison with this system is difficult as this work presented results on the EL task on the union of Web_Manual and Wiki_Links datasets and used the F1 metric. Compared to 83.7 F1 in this work, *TabEL* achieves F1 of 96.92. This is equal to the accuracy of our system on the union of Web_Manual and Wiki_Links as *TabEL* does not ignore any test mention.

5.5 Comparison With Text EL System

EL techniques for free-text input are well established [15, 22–29] and it can be argued that they can be applied to tabular data as well. Here, we evaluate the performance of many existing text EL systems on the Web_Manual-Fixed dataset and show that *TabEL* outperforms all text EL systems on the table EL task. Our results show that the table EL task is better addressed by systems like *TabEL* that are specifically designed to handle tabular data.

Table 3: Macro-averaged precision, recall and F1 score comparison with six text-EL systems on Web_Manual-Fixed dataset. GERBIL link for results: http://gerbil.aksw.org/gerbil/experiment?id=201507180000

	AGDISTIS	**Babelfy**	**Dbpedia**	**KEA**	**NERD-ML**	**WAT**	***TabEL***
	[29]	[30]	**Spotlight**	[31]	[32]	[33]	
			[28]				
Macro-Precision	0.7773	0.9464	0.8248	0.9209	0.7611	0.9490	**0.9855**
Macro-Recall	0.3587	0.3431	0.1086	0.369	0.6907	0.3442	**0.9237**
Macro-F1	0.3835	0.3663	0.1637	0.4008	0.697	0.3695	**0.9237**

We used the GERBIL framework [34] to compare against text EL systems. Each table in the test dataset is converted into text format with their mentions identified and given to the GERBIL framework as input. Table 3 shows the macro-averaged precision, recall and F1 scores of *TabEL* compared with six other text EL systems. *TabEL* significantly outperforms all text EL systems on precision, recall and F1.

5.6 Ablation Study

We performed an ablation study on the TabEL_35K dataset to evaluate the effectiveness of each group of features used in $\mathcal{M}_{LR}$. Table 4 shows the groups of features in descending order of the percentage increase in error when that feature group is removed from $\mathcal{M}_{LR}$. All feature groups included in $\mathcal{M}_{LR}$ have a positive effect on the system, with the SR group of features being the most valuable of all. Context-based features also have a high impact on the accuracy of the overall system.

Table 4: Ablation study: Percentage increase in error, on TABEL_35K dataset, when each group of features is removed from $\mathcal{M}_{LR}$

Feature Group Removed	**Accuracy**	**Percent increase in error**
-SR	95.31	189.28
-Prior	96.33	126.31
-Existing Link Features	96.55	113.08
-Text Context	96.70	103.73
-Entity Context	96.93	89.27
-Surface	98.32	3.54
Full Wikifier	98.38	0.0

6 Analysis

In our experiments above, we show that *TabEL* consistently outperforms previous state-of-the-art systems. One reason for this is that joint approaches, such as the one in [4], make the *common most-specific type (CMST) assumption*: that all else being equal, an EL system should prefer to link mentions within a column to entities sharing a common most-specific type grounded in a KB. In principle, this assumption could be leveraged to prefer disambiguations that resulted in column entities sharing the same data type, and thereby improve accuracy. However, this assumption is often violated in practice because existing KBs, in which the types are grounded, are incomplete and noisy. In fact, when mapped to types in the DBpedia ontology, we find that only 24.3% of the columns from Wikipedia tables satisfy the *CMST* assumption. Further, when a CMST does exist for a column, it is often not specific enough to aid EL: over 50% of the entities in a column remain ambiguous even after restricting entities to the column's CMST. As a result, rather than restricting EL targets based on strict types in a KB, we use a weaker type constraint encoded by features based on entity co-occurrence statistics, i.e. SR and entity context similarity features. At the same time, rather than using a joint model to solve the EL and type identification tasks together, our system solves the EL task for tables in isolation. This allows *TabEL* to sidestep the risks of making the CMST assumption. It has also been found in previous work by Venetis et al. [10] that solving the table column type-identification task in isolation yields better performance than an approach that tackles the three table semantic interpretation tasks jointly.

6.1 Effectiveness of ICA

Table 5 shows the number of iterations it took ICA to converge and the improvement of accuracy due to inference performed in multiple iterations on two test datasets. Results show that collective inference is useful for this task.

	WEB_MANUAL-FIXED	**WIKI_LINKS-RANDOM**
No. of iterations for convergence	6	18
Accuracy after iteration 1	88.50	95.85
Convergence Accuracy	92.94	96.17

Table 5: Effectiveness of ICA on two datasets. On both datasets, accuracy at convergence is higher than the accuracy at the end of the first iteration.

6.2 Analysis of Entity Prevalence Bias

The EL task is known to be easy for prominent entities and particularly difficult for the long-tail of less common entities. Here, we analyze the distribution of prominence of entities in Wikipedia tables and show that our system performs equally well even for the long tail of less prominent entities. On the other hand, accuracy of the $TabEL_{prior}$ system is low for less prominent entities and high for common entities.

We use the number of in-links to an entity in Wikipedia is an indicator of its prominence. Figure 2 (a) shows a histogram of the number of in-links (in log-scale) of the target entity of mentions in the TABEL_35K dataset. Interestingly, the number of in-links of mention targets is log-normally distributed. The estimated normal distribution fit is also shown.

We divide the TABEL_35K dataset into 5 bins at equal intervals of in-link counts (in log scale). Figure 2 (b) shows how the performance of *TabEL* varies as the number of in-links of a mention's target varies. The accuracy of our system remains nearly the same across these bins.

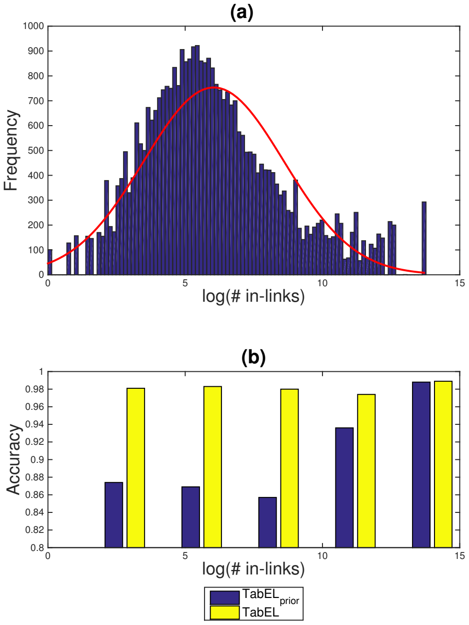

Fig. 2: (a) Histogram of in-link counts (in log-scale) of targets of mentions in Wikipedia tables. A normal distribution fit is shown in red (b) Variation of *TabEL* performance as number of in-links of the target entity varies.

6.3 Run time Analysis

To estimate the scalability of our system, we measured the time taken to disambiguate mentions in each dataset. Table 6 shows the disambiguation time per-table and the number of iterations for collective inference in to converge. Limaye et. al. [4] reported a disambiguation time of 0.7 s / table.

Dataset	Average Time (s/table)	No. of iterations
WEB_MANUAL- FIXED	1.12	6
WIKI_LINKS- RANDOM	2.32	18
WIKI_LINKS	31.9	27

Table 6: Per-table disambiguation time and number of inference iterations of *TabEL* on different datasets

The WIKI_LINKS dataset is densely populated with mentions, hence the higher run-time. There are many parameters, such as number of candidates, maximum number of inference iterations, convergence criteria, that can be tuned to further improve the disambiguation time of our system.

One possible optimization for *TabEL* is the number of entities in the candidate set. We vary a global parameter to threshold the number of candidates for each mention

Table 7: Effect of varying the maximum number of candidates on *TabEL*'s accuracy on the TABEL_35K dataset

Max No. of Candidates	**Average No. of Candidates**	***TabEL* Accuracy**
5	1.68	0.73
10	2.32	0.83
15	2.97	0.90
20	3.54	0.94
25	4.04	0.96
40	7.18	0.98

and analyze its effect on the accuracy of *TabEL* on the TABEL_35K dataset. We find that changing this parameter from 5 to 20 results in a considerable jump in accuracy. Increasing this threshold further gives diminishing returns. Table 7 shows these results along with the average number of candidates per mention as this threshold is varied.

7 Previous Work

Cafarella et al. [1] pioneered the work on Web tables and found that there are 154 million tables on the Web that contain relational data. Since then, various efforts have been made to extract semantics from Web tables. Muñoz et al. [5, 35] described an approach that relies on existing links to convert Wikipedia tables to RDF triples. They use facts from an existing knowledge base (KB) like DBpedia, in order to find existing relations in pairs of columns in tables, and then extract new relations for entities in corresponding columns. Sekhavat et al. [36] proposed a probabilistic approach to augment a KB with facts from tabular data using a Web text corpus and natural language patterns associated with relations in the KB. These methods of RDF extraction which rely on existing links in tables can benefit significantly from our system *TabEL*, which achieves better precision than previous work on the entity linking task on Web tables and performs especially well on Wikipedia tables.

Syed et al. [9] describe methods to automatically infer a partial semantic model of Web tables using Wikitology [37], a topic ontology built using Wikipedia's articles and associated pages. Their system tackles all three tasks of semantic interpretation of tables. Mulwad et al. [6, 7] also jointly model entity linking, column type identification and relation extraction using a graphical model. The closest previous work for our system is by Limaye et al. [4] and Zhang et al. [12]. Both their systems jointly model the entity linking, column class identification and relation extraction tasks for Web tables. As argued in Section 1, owing to heavy reliance on the correctness and completeness of a KB, joint models run the risk of negatively affecting performance on entity linking. Venetis et al. [10] have shown that a system built to handle only the type identification task performs better than the joint model in [4] on this task. In *TabEL*, we focus on the individual task of *entity linking*, and show EL can be similarly improved by solving it in isolation, rather than through a joint approach.

Finally, previous work has studied applications built upon extracted Web tables. Table augmentation has been studied by Das et al. [38], Gupta et al. [21], Fan et al. [39] and our previous work [40]. Das et al. [38] and our previous work [40] also studied *table search*, the task of returning a list of tables for a given text query ranked by their relevance to a text query. All these systems utilize existing entity references in tables in different ways, and adding more links to tables using *TabEL* may improve the accuracy of the applications.

8 Conclusion and Future Work

In this paper, we described our table entity linking system *TabEL*. *TabEL* uses a collective classification technique to collectively disambiguate all mentions in a given table. Instead of using a strict mapping of types and relations into a reference Knowledge Base, *TabEL* uses soft constraints in its graphical model to sidestep errors introduced by an incomplete or noisy KB and outperforms previous work on multiple datasets. We also showed that *TabEL* performs equally well even for the long tail of infrequently-mentioned entities – for which the EL task is particularly hard. Ablation studies demonstrate the effectiveness of our Semantic Relatedness features.

We made our table corpus containing 1.6 million Wikipedia tables publicly available along with annotated datasets which can be used by future table-EL systems for comparison.

In future work, we plan to integrate *TabEL* with systems that identify column types and relations between columns of a table to convert table data into machine-understandable formats like RDF. Finally, we plan to release our code in future.

Acknowledgements

This research was supported in part by NSF grant IIS-1351029 and the Allen Institute for Artificial Intelligence.

References

1. Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549, 2008.
2. Patrice Buche, Juliette Dibie-Barthelemy, Liliana Ibanescu, and Lydie Soler. Fuzzy web data tables integration guided by an ontological and terminological resource. *Knowledge and Data Engineering, IEEE Transactions on*, 25(4):805–819, 2013.
3. Gaëlle Hignette, Patrice Buche, Juliette Dibie-Barthélemy, and Ollivier Haemmerlé. Fuzzy annotation of web data tables driven by a domain ontology. In *The Semantic Web: Research and Applications*, pages 638–653. Springer, 2009.
4. Girija Limaye, S Sarawagi, and S Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB*, pages 1338–1347, 2010.
5. Emir Munoz, Aidan Hogan, and Alessandra Mileo. Triplifying wikipedia's tables. In *LD4IE@ ISWC*, 2013.
6. Varish Mulwad, Tim Finin, and Anupam Joshi. Automatically generating government linked data from tables. In *Working notes of AAAI Fall Symposium on Open Government Knowledge: AI Opportunities and Challenges*, volume 4, 2011.

7. Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In *The Semantic Web–ISWC 2013*, pages 363–378. Springer, 2013.
8. Varish Mulwad, Tim Finin, Zareen Syed, and Anupam Joshi. T2ld: Interpreting and representing tables as linked data. In *9th International Semantic Web Conference ISWC 2010*, page 25. Citeseer, 2010.
9. Zareen Syed, Tim Finin, Varish Mulwad, and Anupam Joshi. Exploiting a web of semantic data for interpreting tables. In *Proceedings of the Second Web Science Conference*, 2010.
10. Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Paşca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment*, 4(9):528–538, 2011.
11. Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. Understanding tables on the web. In *Conceptual Modeling*, pages 141–155. Springer, 2012.
12. Ziqi Zhang. Start small, build complete: Effective and efficient semantic table interpretation using tableminer. *Under transparent review: The Semantic Web Journal*, 2014.
13. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007. ACM Press.
14. Qing Lu and Lise Getoor. Link-based classification. In *ICML*, volume 3, pages 496–503, 2003.
15. Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.
16. Brent Hecht, Samuel H Carton, Mahmood Quaderi, Johannes Schöning, Martin Raubal, Darren Gergle, and Doug Downey. Explanatory semantic relatedness and explicit spatialization for exploratory search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 415–424. ACM, 2012.
17. I Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.
18. Kenneth Ward Church. One term or two? In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 310–318. ACM, 1995.
19. Hannes Dohrn and Dirk Riehle. Design and implementation of the sweble wikitext parser: unlocking the structured data of wikipedia. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 72–81. ACM, 2011.
20. Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.
21. Rahul Gupta and Sunita Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proceedings of the VLDB Endowment*, 2(1):289–300, 2009.
22. Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.
23. Xiao Cheng and Dan Roth. Relational inference for wikification. *Urbana*, 51:61801, 2013.
24. Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2, 2014.
25. Xiao Ling, Sameer Singh, and Daniel S Weld. Context representation for named entity linking. In *Proceedings of the 3rd Pacific Northwest Regional NLP Workshop (NW-NLP'14)*, 2014.

26. Thanapon Noraset, Chandra Bhagavatula, and Doug Downey. Websail wikifier at erd 2014. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 119–124. ACM, 2014.
27. Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716. Citeseer, 2007.
28. Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
29. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. Agdistis-graph-based disambiguation of named entities using linked data. In *The Semantic Web–ISWC 2014*, pages 457–471. Springer, 2014.
30. Andrea Moro, Francesco Cecconi, and Roberto Navigli. Multilingual word sense disambiguation and entity linking for everybody. *Proc. of ISWC (P&D)*, pages 25–28, 2014.
31. Nadine Steinmetz and Harald Sack. Semantic multimedia information retrieval based on contextual descriptions. In *The Semantic Web: Semantics and Big Data*, pages 382–396. Springer, 2013.
32. Marieke Van Erp, Giuseppe Rizzo, and Raphaël Troncy. Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In *# MSM*, pages 27–30. Citeseer, 2013.
33. Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *arXiv preprint arXiv:1006.3498*, 2010.
34. R Usbeck, M Röder, AC Ngonga-Ngomo, C Baron, A Both, M Brümmer, D Ceccarelli, M Cornolti, D Cherix, B Eickmann, et al. Gerbil-general entity annotation benchmark framework. In *24th World Wide Web Conference (WWW)*, 2015.
35. Emir Muñoz, Aidan Hogan, and Alessandra Mileo. Using linked data to mine rdf from wikipedia's tables. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 533–542. ACM, 2014.
36. Yoones A Sekhavat, Francesco di Paolo, Denilson Barbosa, and Paolo Merialdo. Knowledge base augmentation using tabular data. *Linked Data on the Web at WWW2014*, 2014.
37. Zareen Saba Syed, Tim Finin, and Anupam Joshi. Wikitology: Using wikipedia as an ontology. In *proceeding of the second international conference on Weblogs and Social Media*, 2008.
38. Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. Finding related tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 817–828. ACM, 2012.
39. Ju Fan, Meiyu Lu, Beng Chin Ooi, Wang-Chiew Tan, and Meihui Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 976–987. IEEE, 2014.
40. Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Methods for exploring and mining tables on wikipedia. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 18–26. ACM, 2013.